(19) Europäisches Patentamt
European Patent Office
Offic européen des brevets

(11) Publication number: **0 155 211**
A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 85400342.3

(22) Date of filing: 25.02.85

(51) Int. Cl.⁴: G 06 F 9/38

(30) Priority: 24.02.84 JP 34057/84
24.02.84 JP 34056/84

(43) Date of publication of application: 18.09.85
Bulletin 85/38

(84) Designated Contracting States: DE FR GB

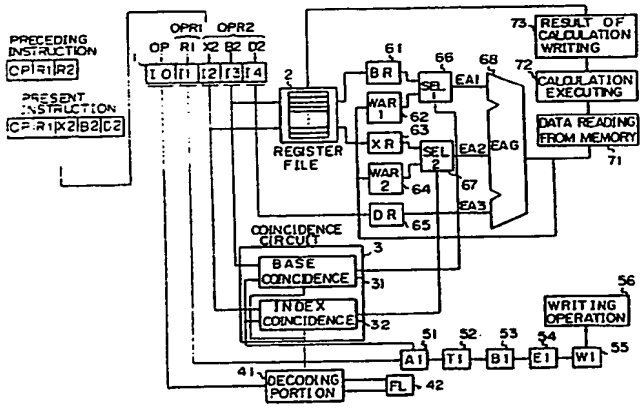
(71) Applicant: FUJITSU LIMITED, 1015, Kamikodanaka Nakahara-ku, Kawasaki-shi Kanagawa 211 (JP)

(72) Inventor: Kitamura, Toshiaki, 2-24-62-2-1008, Tamagawa, Ota-ku Tokyo 146 (JP)
Inventor: Oinaga, Yuji, 4-28-15, Kamiuma, Setagaya-ku Tokyo 154 (JP)
Inventor: Onishi, Katsumi, 2-137, Shimoshingashi, Kawagoe-shi Saitama 356 (JP)

(74) Representative: Descourtieux, Philippe et al, CABINET BEAU de LOMENIE 55 rue d'Amsterdam, F-75008 Paris (FR)

(54) System for by-pass control in pipeline operation of computer.

(57) A system for computer pipeline operation, in which a plurality of instructions are executed in parallel by commencing, before the termination of execution of the preceding instruction, the execution of the present instruction, includes a conflict detection unit (3), a data establishment indication unit (41, 42), and a source data by-pass unit (62, 64, 66, 67). The source data by-pass unit by-passes a source data to the processing stage which requires this source data immediately after conflict is detected between the result data of the preceding instruction and the source data of the present instruction and the establishment of the source data of the present instruction is detected.



EP 0 155 211 A2

0155211

SYSTEM FOR BY-PASS CONTROL IN PIPELINE OPERATION OF COMPUTER

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a system for by-pass control during pipeline operation of a computer.

2. Description of the Related Arts

During a pipeline operation of a computer, a register conflict can occur when a register in which a change of data is under execution through a preceding instruction is referred to by a present instruction. When such a register conflict occurs, reference to this register by the present instruction can be carried out only after completion of the execution of the preceding instruction for a change of data.

Usually the flow of instructions in the pipeline operation consists of the decoding stage D for decoding the instruction, the calculate address stage A for calculating the operand address, the transform address stage T for transforming an operand address into an actual address, the read from buffer stage B for reading an operand from the buffer memory controlled by the storage control portion, the execute stage E for executing the calculation, and the write result stage W for checking the result of the calculation and writing the result.

SUMMARY OF THE INVENTION

It is the object of the present invention to realize an improved pipeline operation of a computer in which a delay in operation due to a register conflict during the flow of the pipeline operation is reduced, and to increase the speed of control of the execution of the instructions in the pipeline operation.

According to the fundamental aspect of the present invention, there is provided a system for by-pass

control during the pipeline operation of a computer, using a plurality of processing stages, in which an instruction is executed in a plurality of periods, and a plurality of instructions are executed in parallel by commencing, before the termination of execution of the preceding instruction, the execution of the present instruction. The system includes a conflict detection unit for detecting the conflict between the result data of the preceding instruction and the source data of the present instruction, a data establishment indication unit for indicating the establishment of the result data of the preceding instruction prior to the usual result writing period, and a source data by-pass unit for by-passing the source data to the processing stage which requires this source data immediately after conflict is detected between the result data of the preceding instruction and the source data of the present instruction and the establishment of the source data of the present instruction is detected.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings, Fig. 1 illustrates examples of the prior art pipeline flow;

Fig. 2 is a schematic block diagram of the system for by-pass control during pipeline operation of a computer according to an embodiment of the present invention;

Fig. 3 illustrates the pipeline flow for explaining the operation of the system shown in Fig. 2;

Fig. 4 illustrates another embodiment of the present invention;

Fig. 5 illustrates the pipeline flow for the operation of the system shown in Fig. 4;

Fig. 6 illustrates the prior art pipeline flow for explaining the difference between that flow and the pipeline flow shown in Fig. 5; and

Fig. 7 illustrates a further embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The system for by-pass control during the pipeline operation of a computer according to an embodiment of the present invention is shown in Fig. 2. The system of Fig. 2 comprises an instruction register 1, a register file 2, a coincidence circuit 3, a decoding portion 41, a by-pass valid flag register 42, register writing address holding registers 51, 52, 53, 54, and 55, and a writing operation portion 56. The system of Fig. 2 also comprises a base register 61, an operand address holding register 62, an index register 63, an operand address holding register 64, a displacement register 65, selectors 66 and 67, an adder/subtracter 68 for calculating addresses, an instruction portion 71 for instructing data reading from a memory, an instruction portion 72 for instructing an execution of calculation, and an instruction portion 73 for instructing the writing of the result of the calculation.

According to the system of the present invention, an instruction fetched from the main storage is registered in the instruction register 1. This instruction consists of, for example, an operation code OP, a first operand OPR1, and a second operand OPR2. Where the instruction is a load type instruction, the first operand OPR1 is an operand to be written, and the second operand OPR2 is an operand for referral. In the example of a stored instruction shown in the instruction register 1 of Fig. 2, the second operand OPR2 comprises an index register number X2, a base register number B2, and a displacement D2.

The register file 2 is a storage circuit consisting of a general use register, a floating point register, and the like. The output of the register file 2 is supplied to the base register 61 and the index register 63.

The selector 66 receives the outputs of the base register 61 and the operand address holding register 62 as inputs to be selected and the output of coincidence

circuit 3 as a control input, and the selector 67 receives the outputs of the index register 63 and the operand address holding register 64 as inputs to be selected and the output of coincidence circuit 3 as a control input.

The adder/subtracter 68 receives the data from the selectors 66 and 67, and the displacement register 65, and carries out addition or subtraction on the basis of the received data to obtain an operand address.

The operand address calculated by the adder/subtracter 68 is supplied to the registers 62 and 64 for holding the operand address.

In the prior art, the registers for holding operand addresses, such as the registers 62 and 64 in Fig. 2, are provided for temporarily holding and updating an operand address in order to facilitate the treating of an operand address having a variable length in the main storage. In the system of Fig. 2 according to the present invention, these registers are utilized for holding by-pass data.

The coincidence circuit 3 detects the identification between a register which stores the result of the execution of the preceding instruction and a register to be used for calculating the operand address of the present instruction. The selections in the selectors 66 and 67 are controlled by the outputs of the coincidence circuit 3.

The decoding portion 41 is for analyzing the operation code of an instruction. The decoding portion 41 decides whether or not the preceding instruction is an instruction which can realize the execution of the calculation of the register writing data, on the basis of the output of the adder/subtracter 68 for calculating addresses. The valid flag register 42 is set or reset on the basis of this decision.

For example, when the instruction can obtain the result of the execution of the instruction at a rela-

tively early cycle of the execution of the instruction, such as a load-address (LA) instruction, the valid flag is set in the valid flag register 42. When the valid flag is set, the by-pass control enable signal is supplied from the decoding portion 41 to the coincidence circuit 3.

The register writing addresses are stored as follows: A1 for stage A in register 51; T1 for stage T in register 52; B1 for stage B in register 53; E1 for stage E in register 54; and W1 for stage W in register 55.

The coincidence circuit 3 consists of a base coincidence circuit 31 and an index coincidence circuit 32. The base coincidence circuit 31 detects the coincidence between the register writing address A1 of the preceding instruction, such as that of the load-address instruction and the base modification register address, shown as the I3 portion of the present instruction. When coincidence is detected, the data of the register 62, (not the data read from the register file 2), is supplied to the input port EA1 of the adder/subtracter 68 to carry out the by-pass operation.

The index coincidence circuit 32 detects the coincidence between the register writing address A1 of the preceding instruction and the index modification register address, shown as the I2 portion of the present instruction. When coincidence is detected, the data of the register 64, (not the data read from the register file 2), is supplied to the input port EA2 of the adder/subtracter 68 to carry out the by-pass operation.

Thus, in the system of Fig. 2, a pipeline flow having a minimum delay, as shown in Fig. 3, (B), is realized. The time required for the operation shown in Fig. 3, (B) is reduced by 5 cycles less than that shown in Fig. 1, (B), and by 4 cycles less than that shown in Fig. 1, (C).

Even if the register writing address A1 coincides

with the base or the index, the by-pass operation is not carried out when the valid flag is reset in the valid flag register 42, and the operation is carried out with the usual waiting time.

Although in the above description the load-address (LA) instruction is adopted as an instruction which can instruct the calculation of the register writing data on the basis of the output of the adder/subtracter 68 for calculating addresses, other instructions can be adopted provided that the adopted instruction can obtain the result of the execution of the instruction at an early cycle of the execution of the instruction.

The system for by-pass control in the pipeline operation of a computer according to another embodiment of the present invention is shown in Fig. 4. The system of Fig. 4 comprises an instruction register 1, a register file 2, a coincidence circuit 3, a decoding portion 41, by-pass valid flag registers 421, 422, 423, 424, and 425, and register writing address holding registers 51, 52, 53, 54, and 55.

The system of Fig. 4 also comprises a base register 61, an index register 63, selectors 661, 662, 671, and 672, a displacement register 65, and an adder/subtracter 68 for calculating addresses. The system of Fig. 4 further comprises by-pass registers 811, 812, 813, and 814 corresponding to the stages of the operation, selectors 821, 822, 823, and 824, a result of calculation storing portion 741, a calculation executing portion 742, and a data reading from memory portion 743.

The adder/subtracter 68 for calculating addresses carries out addition or subtraction on the basis of the data from the base register 61, the index register 63, the displacement register 65, and the by-pass registers 811, 812, 813, and 814 for holding an operand address, and calculating an operand address. The operand address calculated by the adder/subtracter 68 is used for a memory address and is supplied successively to the

by-pass registers 811, 812, 813, and 814. The number of by-pass registers corresponds to the number of stages of the pipeline after stage A.

The coincidence circuit 3 comprises coincidence elements 311 to 314 and 321 to 324, and NOR gates 331 and 341. The coincidence elements carry out the comparison between the base modification portion I3 or the index modification I2 of the instruction, and the register writing addresses A1 to E1 in the stages of the operation.

For example, the coincidence element 311 detects the coincidence between the register writing address A1 of the preceding instruction, such as a load address (LA) instruction, and the base modification address I3 of the present instruction. If coincidence is detected, the data of the by-pass register 811 (TBYR) is supplied to the input port EA1 of the adder/subtracter 68 to cause the by-pass operation.

If no coincidence is detected in the coincidence elements 311 to 314, the data of the base register 61 is supplied to the input port EA1 of the adder/subtracter 68.

Thus, in the system of Fig. 4, a pipeline flow having a minimum of delay, as shown in Fig. 5, is realized.

To illustrate the difference between the embodiment of the present invention and the prior art, the pipeline flow for the prior art is shown in Fig. 6.

A further embodiment of the present invention is illustrated in Fig. 7. The system shown in Fig. 7 comprises selectors 911, 912, and 913, by-pass registers 921, 922, and 923, and identification number registers 931, 932, and 933. The control circuit including the coincidence circuit and decoding portion, such as shown in Fig. 4, is denoted as CC.

The identification numbers (ID) 0, 1, and 2 are assigned to the by-pass registers 921, 922, and 923,

respectively, for the read from buffer stage B, the execute stage E, and the write result stage W. The transfer of the data from the by-pass register 811 (TBYR) to the registers 921, 922, and 923 is carried out by changing the identification number successively with the sequence 0, 1, 2, 0, 1, 2, ...

Only the identification numbers are held in the identification number registers 931, 932, and 933. These identification numbers are used for selecting registers where the by-pass control is carried out with reference to the data in the identification number registers.

In the system shown in Fig. 7, when a register conflict occurs, a by-pass operation of the data of the by-pass register as a register read data and the operand address calculation are carried out, provided that the corresponding by-pass valid flags in the by-pass valid flag registers 422 to 425 indicate the valid state and that the data is registered in the by-pass register, thus enabling the pipeline operation to proceed satisfactorily.

CLAIMS

1. A system for by-pass control in pipeline operation of a computer, using a plurality of processing stages, an instruction being executed in a plurality of periods, a plurality of instructions being executed in parallel by commencing, before the termination of execution of the preceding instruction, the execution of the present instruction, said system comprising:

a conflict detection means for detecting conflict between the result data of the preceding instruction and the source data of the present instruction,

a data establishment indication means for indicating the establishment of the result data of the preceding instruction prior to the usual result writing period, and

a source data by-pass means for by-passing a source data to the processing stage which requires this source data immediately after conflict is detected between the result data of the preceding instruction and the source data of the present instruction and the establishment of the source data of the present instruction is detected.

2. A system according to claim 1, wherein said processing stages include an operand address calculation stage, an operand data fetch stage, a calculation execution stage, and a result writing stage, and said conflict detection means detects conflict between the register into which the result of the execution of the preceding instruction is to be written and the register from which the data necessary for the address calculation for the present instruction is to be read.

3. A system according to claim 1, wherein said data establishment indication means indicates the fact that, in a specific instruction, the data to be written in the result writing execution period is the data which is established in the address calculation execution

period.

4. A system according to claim 1, wherein said process stages include an operand address calculation circuit for calculating an operand address; and an operand address holding circuit for holding said calculated operand address;

the control being carried out in such a manner that, when a register conflict occurs concerning the input data used for the address calculation in pipeline operation of a computer, under the condition that the preceding instruction is an instruction which can define the content of the data to be written into the register by a calculation circuit for calculating the operand address, the data for calculation of the operand address for the present instruction is supplied via a by-pass from the operand address holding circuit.

5. A system according to claim 1, wherein said processing stages include an operand address calculation circuit for calculating an operand address; a plurality of by-pass registers, the number of said by-pass registers being dependent on the number of the pipeline stages to which the result of the calculation in said operand address calculation circuit are registered; a coincidence circuit for detecting the register conflicts in the pipeline operation; and a by-pass permission data storage circuit for storing by-pass permission data in accordance with the kinds of instruction in correspondence with each of said by-pass registers;

the control being carried out in such a manner that, when a register conflict occurs concerning the input data used for the address calculation in pipeline operation of a computer, under the condition that the preceding instruction is an instruction which can define the content of the data written into the register by the data from said operand address calculation circuit or the data read from a storage portion, the data for base modification or index modification in

the operand address calculation for the present instruction is supplied via a by-pass from said by-pass registers.

0155211

Fig. 1 A FLOW OF INSTRUCTIONS

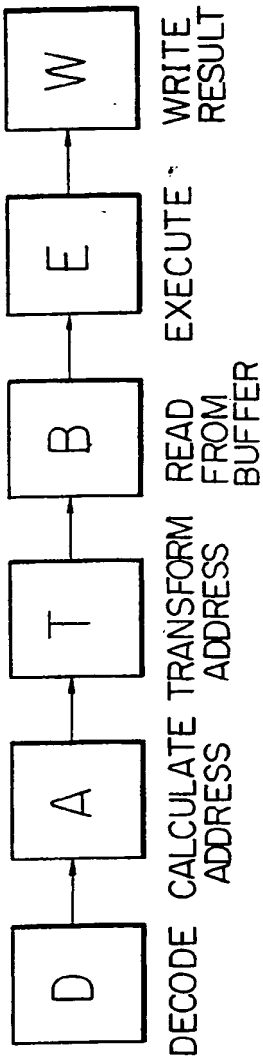


Fig. 1 B EXAMPLE OF PIPELINE FLOW

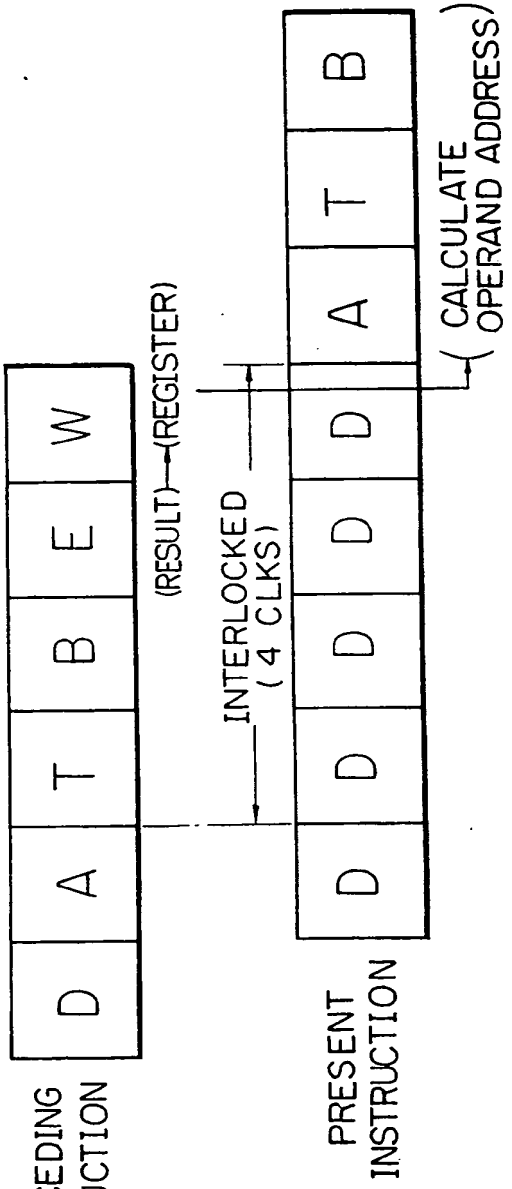
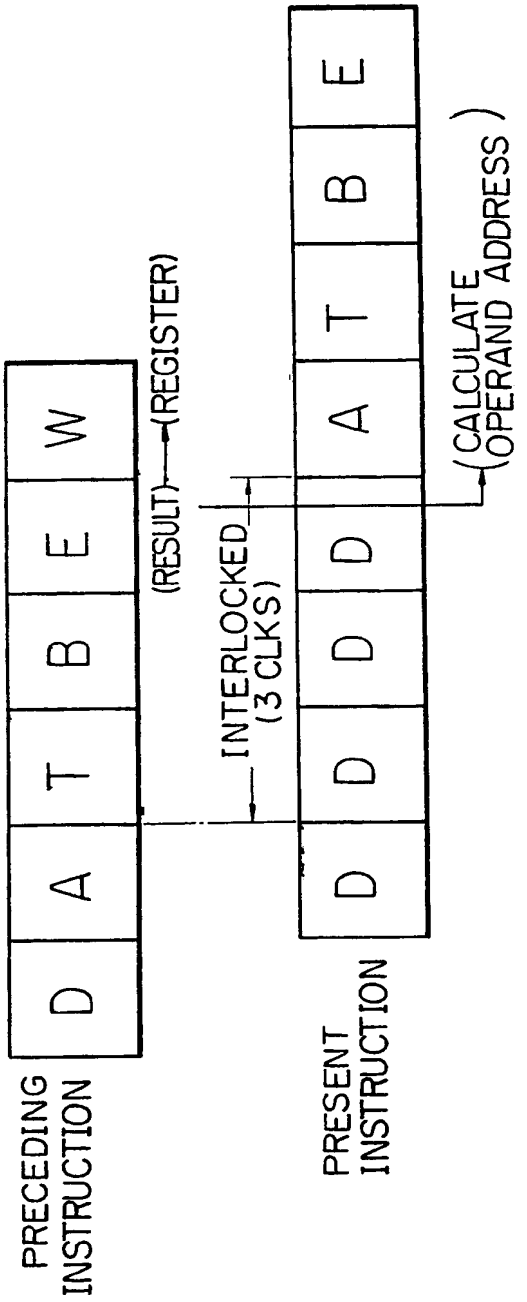


Fig. 1C EXAMPLE OF PIPELINE FLOW



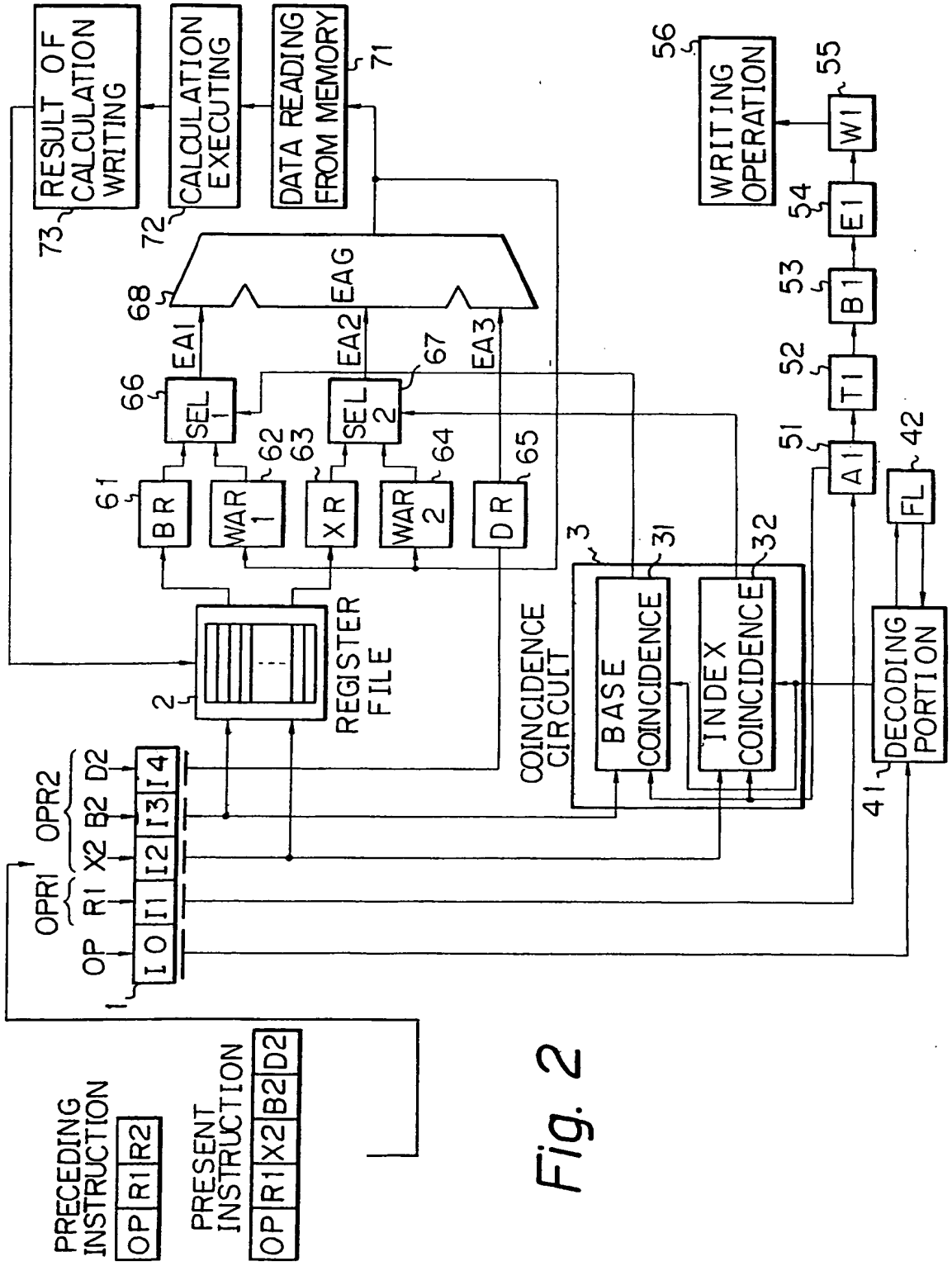

RESULT OF CALCULATION WRITING
CALCULATION EXECUTING
DATA READING FROM MEMORY
WRITING OPERATION
EAG
EA1
EA2
EA3
SEL 1
SEL 2
BR
WAR 1
XR
WAR 2
DR
A1
T1
B1
E1
W1
FL
REGISTER FILE
COINCIDENCE CIRCUIT
BASE COINCIDENCE
INDEX COINCIDENCE
DECODING PORTION
OPR1
OPR2
OP
R1
X2
B2
D2
I0
I1
I2
I3
I4
PRECEDING INSTRUCTION
PRESENT INSTRUCTION
R2

Fig. 2

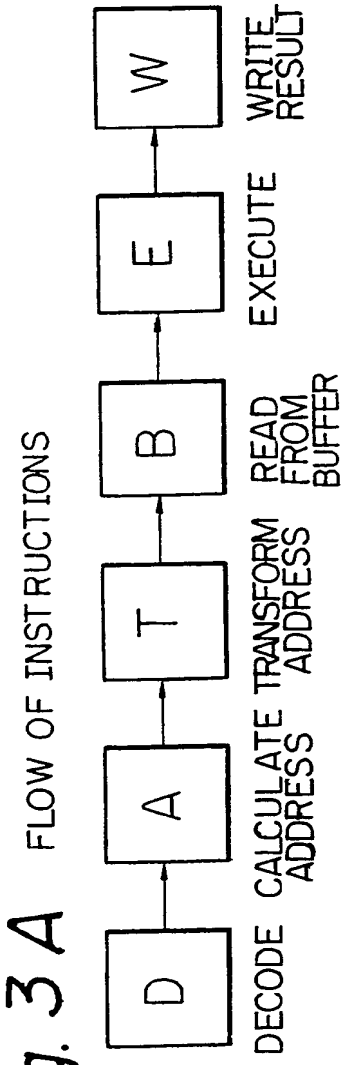


Fig. 3 A FLOW OF INSTRUCTIONS

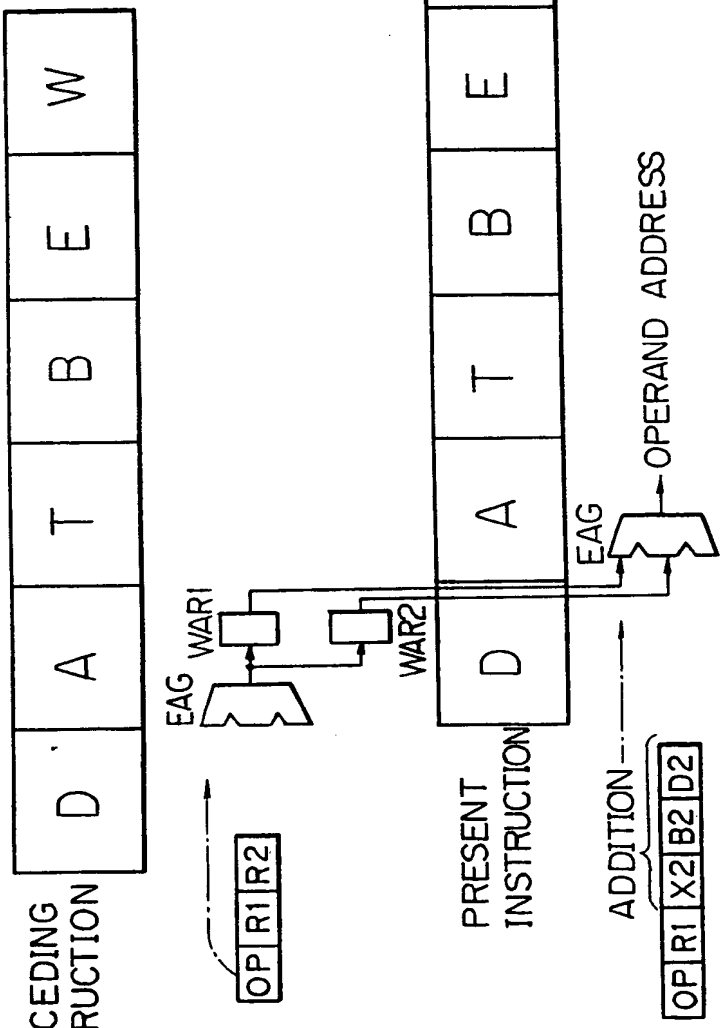


Fig. 3 B PIPELINE FLOW

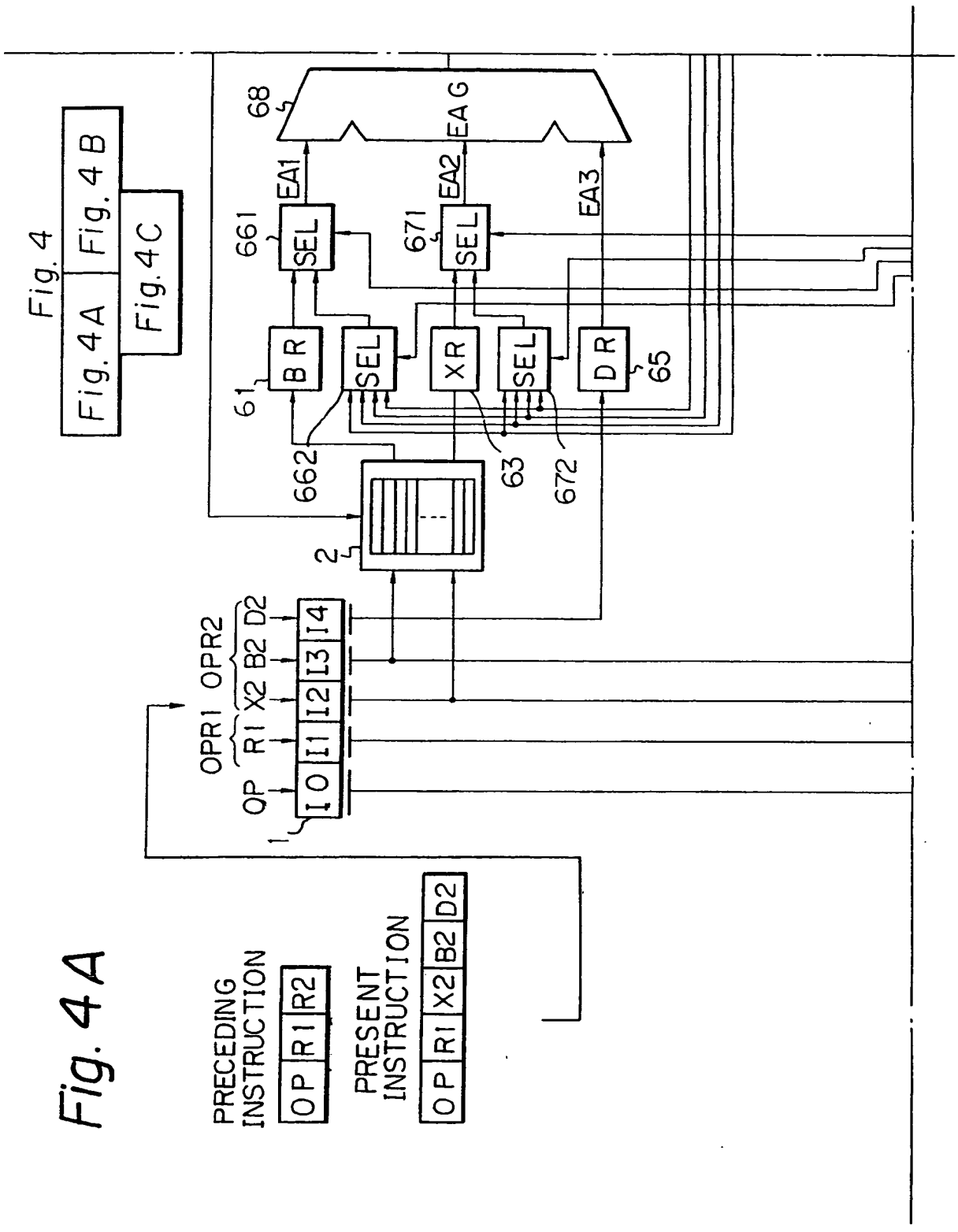


Fig. 4
Fig. 4A
Fig. 4B
Fig. 4C
68
EAG
EA1
EA2
EA3
661
671
SEL
SEL
SEL
SEL
61
BR
XR
DR
65
662
63
672
2
OPR1
OPR2
OP
R1
X2
B2
D2
I0
I1
I2
I3
I4
1
PRECEDING INSTRUCTION
OP
R1
R2
PRESENT INSTRUCTION
OP
R1
X2
B2
D2

Fig. 4A

Fig. 4B

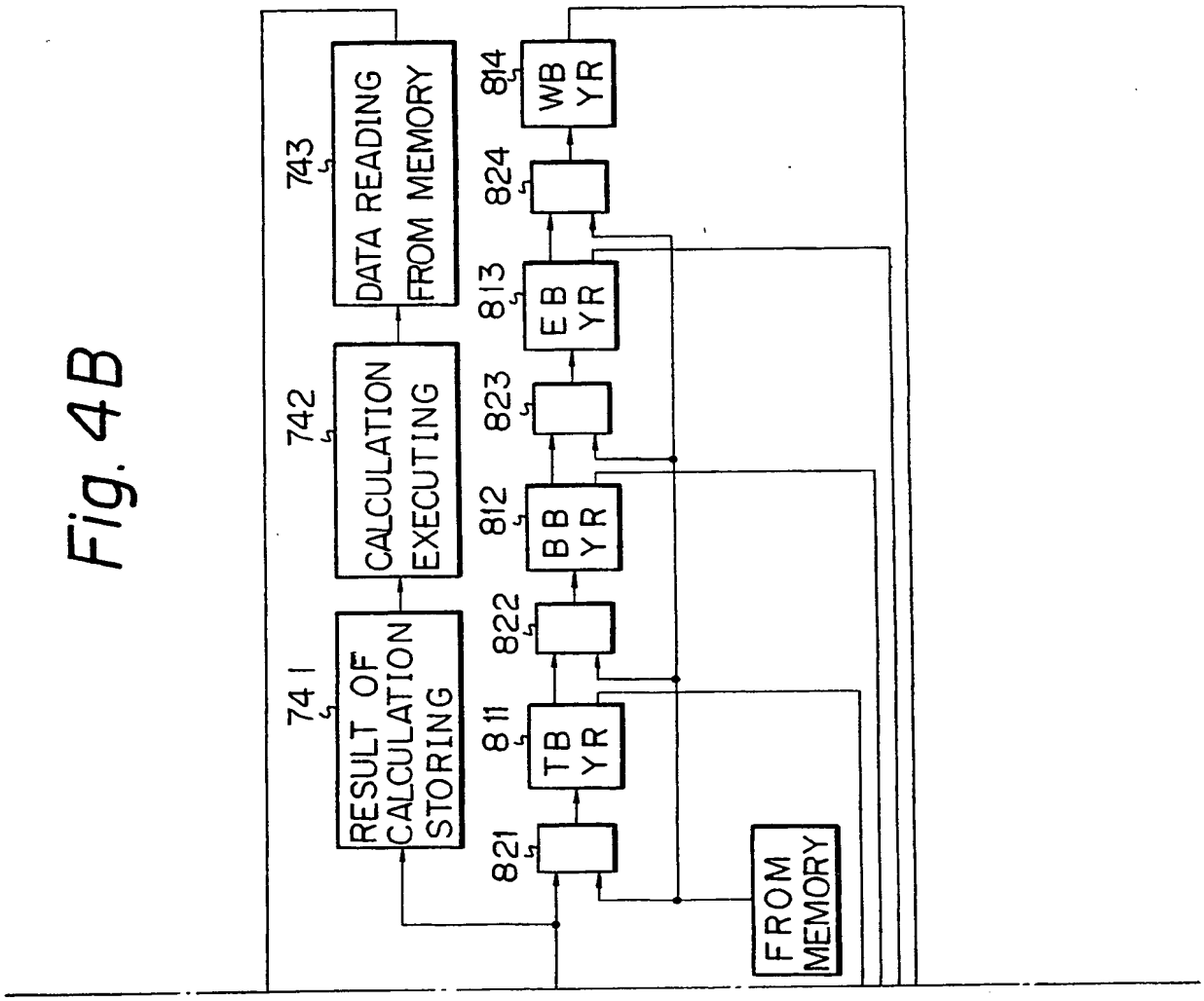


Fig. 4C

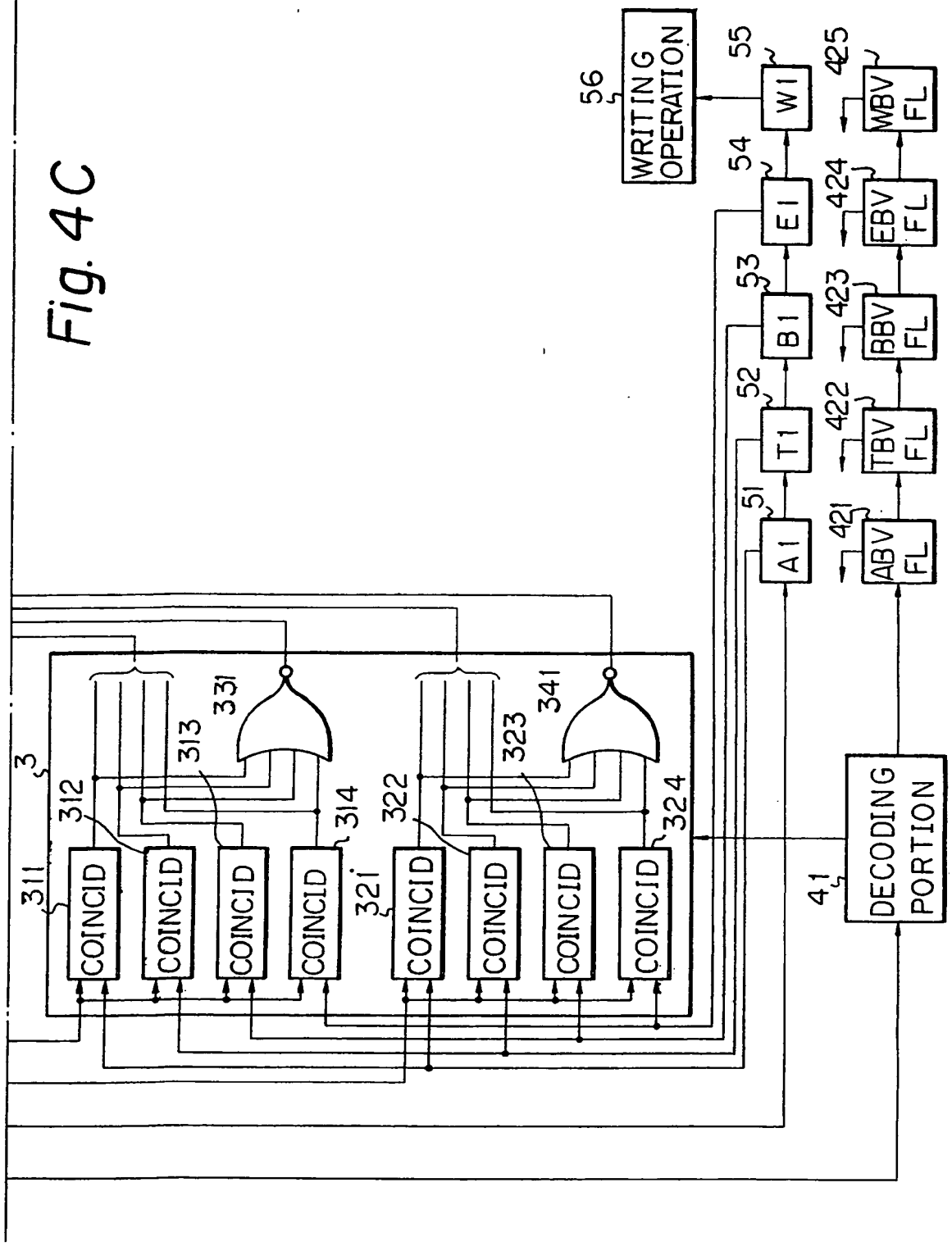

WRITING OPERATION
56
55
W1
425
MBV FL
54
E1
424
EBV FL
53
B1
423
BBV FL
52
T1
422
TBV FL
51
A1
421
ABV FL
331
341
313
323
3
312
322
314
324
311
321
COINCID
41
DECODING PORTION

Fig. 5 PIPELINE FLOW FOR SYSTEM OF FIGURE 4

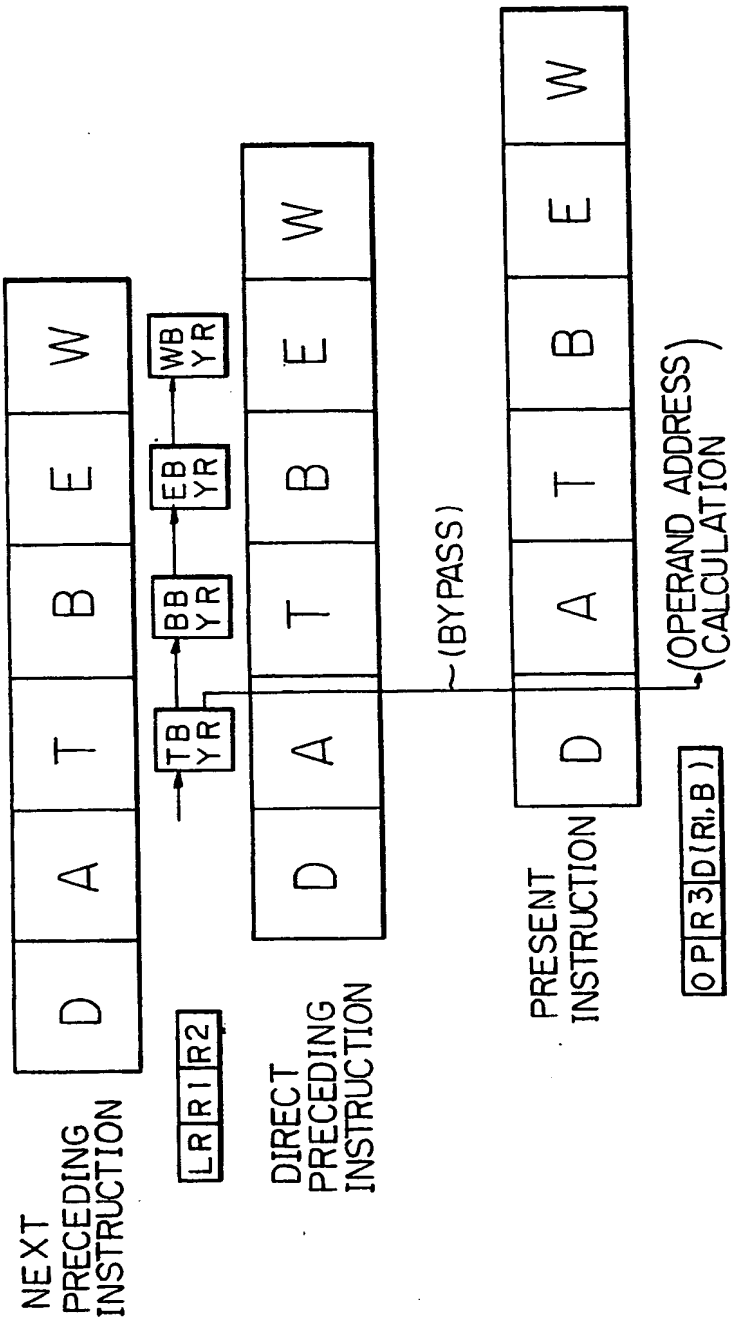


Fig. 6 PIPELINE FLOW FOR PRIOR ART

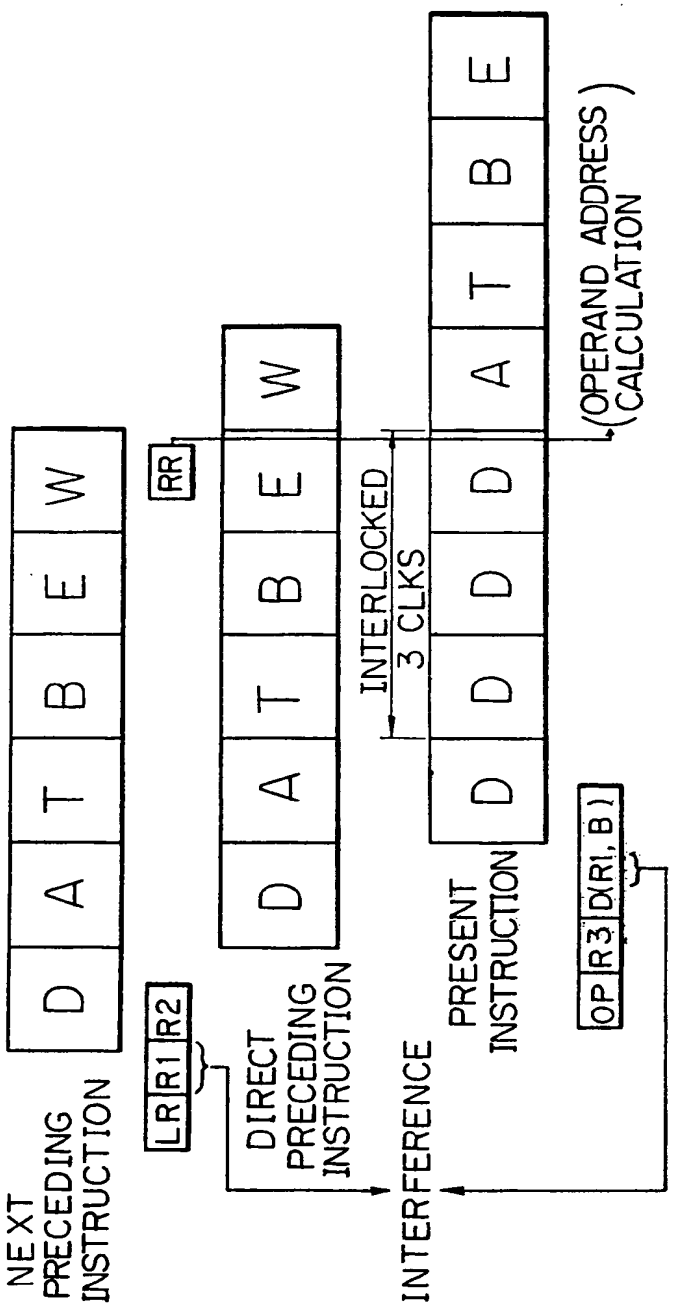
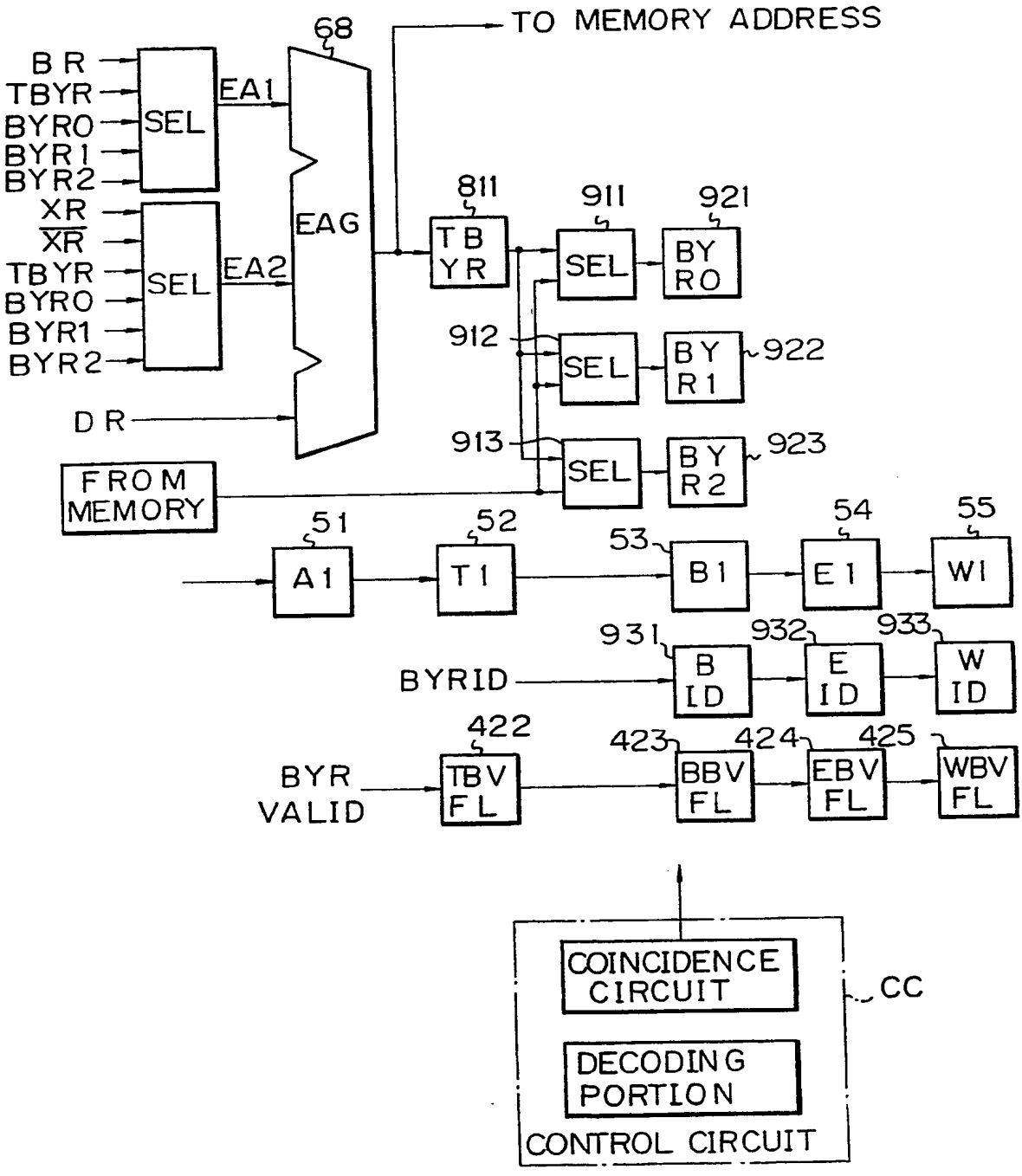


Fig. 7


68
TO MEMORY ADDRESS
BR
TBYR
BYRO
BYR1
BYR2
SEL
EA1
XR
XR
TBYR
BYRO
BYR1
BYR2
SEL
EA2
EAG
DR
FROM MEMORY
811
TB YR
911
921
SEL
BY RO
912
SEL
BY R1
922
913
SEL
BY R2
923
51
A1
52
T1
53
B1
54
E1
55
W1
931
B ID
932
E ID
933
W ID
BYRID
422
TBV FL
423
BBV FL
424
EBV FL
425
WBV FL
BYR VALID
COINCIDENCE CIRCUIT
CC
DECODING PORTION
CONTROL CIRCUIT

(19) Eur päisches Patentamt
European Patent Office
Offic européen des brevets

(11) Publication number: **0 155 211**
A3

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: **85400342.3**

(51) Int.Cl.³: **G 06 F 9/38**

(22) Date of filing: **25.02.85**

(30) Priority: **24.02.84 JP 34057/84**
24.02.84 JP 34056/84

(43) Date of publication of application:
18.09.85 Bulletin 85/38

(88) Date of deferred publication of search report: **27.04.88**

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **FUJITSU LIMITED**
1015, Kamikodanaka Nakahara-ku
Kawasaki-shi Kanagawa 211(JP)

(72) Inventor: **Kitamura, Toshiaki**
2-24-62-2-1008, Tamagawa
Ota-ku Tokyo 146(JP)

(72) Inventor: **Oinaga, Yuji**
4-28-15, Kamiuma
Setagaya-ku Tokyo 154(JP)

(72) Inventor: **Onishi, Katsumi**
2-137, Shimoshingashi
Kawagoe-shi Saitama 356(JP)

(74) Representative: **Descourtieux, Philippe et al,**
CABINET BEAU de LOMENIE 55 rue d'Amsterdam
F-75008 Paris(FR)

(54) **System for by-pass control in pipeline operation of computer.**

(57) A system for computer pipeline operation, in which a plurality of instructions are executed in parallel by commencing, before the termination of execution of the preceding instruction, the execution of the present instruction, includes a conflict detection unit (3), a data establishment indication unit (41, 42), and a source data by-pass unit (62, 64, 66, 67). The source data by-pass unit bypasses a source data to the processing stage which requires this source data immediately after conflict is detected between the result data of the preceding instruction and the source data of the present instruction and the establishment of the source data of the present instruction is detected.

EP 0 155 211 A3

./...

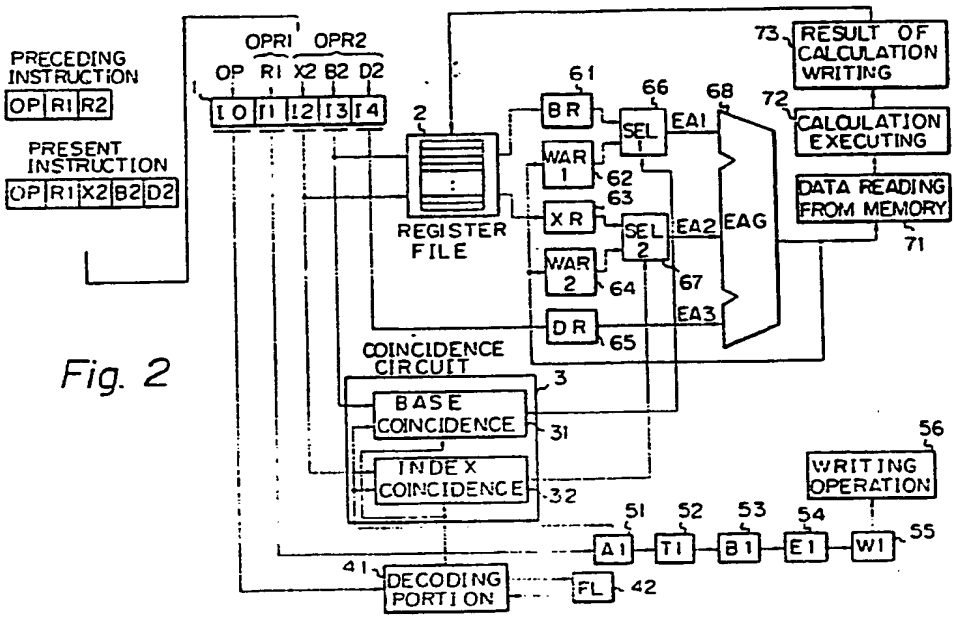

PRECEDING INSTRUCTION
OP R1 R2
PRESENT INSTRUCTION
OP R1 X2 B2 D2
OPR1
OPR2
OP R1 X2 B2 D2
I 0 I 1 I 2 I 3 I 4
1
2
REGISTER FILE
61
BR
WAR 1
62
63
XR
WAR 2
64
65
DR
66
SEL 1
SEL 2
67
68
EA1
EA2
EA3
EAG
73
RESULT OF CALCULATION WRITING
72
CALCULATION EXECUTING
DATA READING FROM MEMORY
71
COINCIDENCE CIRCUIT
3
BASE COINCIDENCE
31
INDEX COINCIDENCE
32
56
WRITING OPERATION
51
52
53
54
55
A1
T1
B1
E1
W1
41
DECODING PORTION
FL
42

Fig. 2

0155211

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 85 40 0342

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
E	EP-A-0 134 620 (PRIME COMPUTER) * Figures 8,12; claims 30-32 * ---	1,2,4	G 06 F 9/38
E	EP-A-0 147 775 (HITACHI) * Figure 1; page 3, lines 1-14 * ---	1,2,4	
E	EP-A-0 147 684 (NEC) * Figure 3; page 7, line 18 - page 11, line 14 * ---	1,2,4	
A	GB-A-2 115 964 (HITACHI) * Page 3, line 36 - page 4, line 13 * ---	1,2,4	
A	EP-A-0 097 956 (HITACHI) * Page 17, line 14 - page 19, line 19; page 25, line 12 - page 26, line 1 * ---	1,2,4	
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 14, no. 3, August 1971, pages 868,869, New York, US; R.J. BULLIONS et al.: "Resolving store-load links in an instruction unit" * Whole article * ---	1,2,4	TECHNICAL FIELDS SEARCHED (Int. Cl.4) G 06 F 9/38
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 14, no. 10, March 1972, pages 2930-2933, New York, US; D. SOFER et al.: "Parallel pipeline organization of execution unit" * Page 2932, paragraph k * -----	1,2,4	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	08-01-1988	QUESSON C.J.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)